

CRM-järjestelmän käyttöliittymän suunnittelu ja toteutus

Pipsa Paatsio

27.5.2016



Tekijä(t) Pipsa Paatsio	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Raportin/Opinnäytetyön nimi CRM –järjestelmän käyttöliittymän toteutus	Sivu- ja liitesivumäärä 19
Opinnäytetyön nimi englanniksi CRM – UI, design and development	
<p>Opinnäytetyönä toteutettiin CRM-järjestelmä HeiHei Study & Travel -yritykselle. Järjestelmä toteutettiin parityönä, josta tämä opinnäytetyö keskittyy käyttöliittymän suunnittelun ja toteutuksen kuvaamiseen.</p> <p>Opinnäytetyön toiminnallisessa osuudessa kehitettiin CRM-järjestelmä PHP-kielellä, AngularJS:ää, JSONia ja Bootstrapia hyödyntäen. Kehitystyö toteutettiin ketteränä ohjelmistokehityksenä Scrum-projektinhallintamenetelmän periaatteita soveltaen. Tämä opinnäytetyöraportti on CRM-järjestelmän teoriaosuus, jossa käydään läpi käytetyt tekniikat sekä niiden hyödyntäminen käyttöliittymän suunnittelussa ja toteutuksessa.</p> <p>Valmiissa produktissa on omat näkymänsä sekä asiakkaille että HeiHei Study & Travelin työntekijöille. Opiskelijat lisäävät tietonsa järjestelmään rekisteröitymisen yhteydessä, ja myöhemmin he voivat tarkastella oman hakuprosessinsa etenemistä järjestelmän kautta. HeiHei Travelin työntekijöillä on laajemmat käyttöoikeudet ja he voivat selata opiskelijoita korkeakoulukohtaisesti, sekä tarkastella missä vaiheessa hakuprosessia kulloinkin mennään.</p> <p>Toiminnallisen ja teoriaosuuden pohjalta havainnoitiin huomioon otettavia asioita käyttöliittymäsuunnittelussa valittuja tekniikoita käyttäen sekä pohdittiin pienyrityksen ja opinnäytetyötiimin välisen yhteistyön etuja ja ongelmakohtia.</p>	
Asiasanat AngularJs, käyttöliittymä, CRM, PHP, Twitter Bootstrap, JSON	

Author(s) Pipsa Paatsio	
Degree programme Business Information Technology	
Report/thesis title CRM – UI, design and development	Number of pages and appendix pages 19
<p>The objective of this thesis was to design and implement a CRM system for a company called HeiHei Study & Travel. The system was created by a team of two programmers, of whom one was responsible for front-end and another for back-end. This report is the written part of the thesis, and summarises the front end part namely. the design and implementation of the user interface and its functions.</p> <p>The product completed through this process was developed using the PHP programming language. The other techniques used were AngularJS, JSON and Bootstrap. Agile software development practices were also a part of this process, by means of Scrum practices. The theoretical part of the thesis describes all the techniques, and how they were exploited during the design and development process.</p> <p>The finished products user interface has its own dashboards for both the customers and the employees of HeiHei Study & Travel. During the registration process the customers add their personal information to the system, and later on the system gives them the possibility to view the details of their personal information and study applications. The employees of HeiHei Study & Travel have extended privileges and they can view the details of each customer and their application processes.</p> <p>On the basis of the theory and software development process, observations were made what to bear in mind regarding UI design, when using the above mentioned techniques. The pros and cons of the cooperation between the commissioner and the thesis project team were also reflected upon and taken into consideration.</p>	
Keywords The abstract ends with a list of keywords, 3–6 words that best describe the contents of your thesis. Place the words in order of importance.	

Sisällys

1	Johdanto	1
2	Lista käytetyistä termeistä ja lyhenteistä	2
3	Tietoperusta	3
3.1	SPA	3
3.2	AngularJs.....	3
3.3	Bootstrap	4
3.4	JSON	4
3.5	Ketterä ohjelmistokehitys ja Scrum	5
3.6	Henkilötietojen käsittely.....	5
3.6.1	Henkilötietojen siirrosta ulkomaille	6
4	CRM-järjestelmän toteutus	7
4.1	Järjestelmän rajoitusten kartoitus	7
4.2	Tietotarpeiden kerääminen	8
4.3	Käyttöliittymän suunnittelu	8
4.4	Selainsovelluksen rakenne ja toteutus	8
5	Käyttäjäpolut	10
5.1.1	Admin-käyttäjän polku	10
5.1.2	Opiskelijakäyttäjän polku.....	13
5.2	AngularJS –integraatio.....	16
5.3	Uudelleenmäärittely ja muutosten toteutus.....	16
6	Yhteenveto	17
	Lähteet	19

1 Johdanto

Tämän opinnäytetyön aiheena on toteuttaa helppokäyttöinen SPA-mallia toteuttava CRM-järjestelmä asiakasyrityksen käyttöön. Järjestelmä toteutettiin parityönä sovelluskehityksen jakautuessa kahteen eri osuuteen kehittäjien kesken; taustajärjestelmän sovelluskehitykseen sekä selainjärjestelmän sovelluskehitykseen. Tässä opinnäytetyöraportissa keskitytään käsittelemään selainjärjestelmän sovelluskehitystä, eli järjestelmän käyttöliittymän toteutusta ja sen integraatiota taustajärjestelmän kanssa.

Sovelluskehitystyössä ohjelmointikieleksi valittiin PHP asiakkaan käyttämän palvelimen asettamista rajoituksista johtuen. Tietokannasta haetut tiedot välitetään selaimelle JSON-objektina ja ne näytetään web-sivuilla AngularJS:ää hyödyntäen. Ulkoasun suunnittelussa ja toteutuksessa hyödynnettiin Twitter Bootstrap –valmiskirjaston tarjoamia komponentteja. Projektinhallinnallisiin asioihin sovellettiin Scrumin tekniikoita ja periaatteita. Scrumia päätettiin olla käyttämättä laajamittaisesti sovelluskehitystiimin pienuudesta johtuen.

Asiakasyrityksenä on HeiHei Study & Travel, jonka pääasiallinen liiketoiminta keskittyy vaihto-opiskelijapalveluihin Suomessa ja ulkomailla. Työn päämääränä on helpottaa vaihto-opintoihin hakeutuvien opiskelijoiden hakuprosessia tarjoamalla yksi keskitetty paikka, johon kaikki tarvittavat liitteet tulee toimittaa ja joka tarpeen tullen tarjoaa apua hakuprosessiin liittyvissä asioissa (HeiHei! Study, Work & Travel, 2016).

Opinnäytetyönä toteutettava järjestelmä tulee HeiHei Study & Travel:in sekä sen asiakkaiden käyttöön. Järjestelmän tarkoituksena on keskittää yrityksen ydinliiketoiminta; jatkossa kaikki asiakastiedot löytyvät samasta paikasta, samaten kuin kaikki asiakkaiden toimittamat asiakirjat ja liitteet. Lähtökohtaisesti asiakastiedot olivat useassa eri paikassa (sähköpostissa, paperilla, koneella eri kansioissa jne.), mikä koettiin haasteeksi. Uuden CRM-järjestelmän myötä sekä asiakkaat että HeiHei Study & Travelin työntekijät pystyvät lisäämään, poistamaan ja hallinnoimaan asiakas-, haku-, ja koulutietoja keskitetysti.

2 Lista käytetyistä termeistä ja lyhenteistä

AngularJs	AngularJS JavaScript-kehikko
Bootstrap –elementti	Bootstrap –kirjaston tarjoama valmis web-sovelluskehitys elementti muotoiluineen, esimerkiksi navigointipalkki, lomake jne.
CRM	Customer Relationship Manager
CSS	Cascading Style Sheets
end-to-end	Taustajärjestelmän ja selainjärjestelmän välinen tiedonsiirto
HTML	HyperText Markup Language
HTML5	HyperText Markup Language Version 5
JS	JavaScript
JSON	JavaScript Object Notation
PHP	Hypertext Preprocessor
REST	Representational State Transfer
SPA	Single Page Application
tagi	ohjelmistokehyksissä esiintyvä merkintätapa, esimerkiksi AngularJS:n käyttämä ng-etuliite
XML	Extensible Markup Language

3 Tietoperusta

Seuraavassa kappaleessa kuvataan käyttöliittymän toteutuksessa käytetty termistö ja menetelmät.

3.1 SPA

SPA-mallissa selainosuus ja taustajärjestelmäosuus nähdään erillisinä järjestelminä, joissa tiedot haetaan taustajärjestelmän staattisista resursseista ja esitetään dynaamisesti muuttuvana sisältönä selaimessa. Tarkoituksena on, että tietty web-sivu ladataan vain kerran, ja sen sisältö generoidaan käyttäjäkohtaisesti tämän tekemien valintojen perusteella. Tämä tekee sovelluksen käytöstä miellyttävämpää, nopeampaa ja responsiivisempaa (CodeProject, 2016).

Perinteisiin web-sovelluksiin verrattuna SPA-sovelluksen luominen on kuitenkin haastavampaa, koska sivujenvälinen tiedonsiirto toteutetaan pääasiassa JavaScriptin avulla. SPA –sovelluksen moitteeton toiminta vaatii lisäksi myös sen, että käyttäjän käyttämä selainohjelma tukee JavaScriptia. Vaikka sovelluksen pitkäaikainen käyttö on nopeampaa kuin perinteisen web-sovelluksen, ensimmäinen sivunlataus voi olla hitaampaa johtuen ladattavien resurssien suuresta määrästä (Tutorials4Geeks, 2016).

3.2 AngularJs

AngularJS on selainpohjaisissa sovelluksissa käytetty avoimen lähdekoodin JavaScriptiä hyödyntävä ohjelmistokehys. Sen avulla muuten staattisille web-sivuille saadaan dynaamisuutta ja muuttuvaa sisältöä. Misko Hevery ja Adam Ambrosin olivat alkuperäiset kehittäjät, jotka aloittivat AngularJS:n kehitystyön oman työnsä sivuprojektina vuonna 2009. Projektin tarkoituksena oli luoda end-to-end –työkalu, joka mahdollistaisi tiedonvaihdon selainjärjestelmän ja taustajärjestelmän välillä (Austin, A., 2014). Myöhemmin AngularJS siirtyi Googlen jatkokehitykseen, ja sen ensimmäinen versio julkaistiin GitHubissa vuonna 2010 (GitHub, 2016).

AngularJS:n käyttöönotto tapahtuu lataamalla AngularJS ohjelmistokehityksen sovellukseen, joka käytännössä toteutetaan lisäämällä web-sivulle AngularJS script –tagit. Tämän jälkeen html-elementteihin voidaan upottaa elementtiin sidotut AngularJs – attribuutit, joiden avulla haluttu sisältö haetaan AngularJS-controllerista (AngularJS, 2016).

Html –elementteihin sijoitetut angular-tagit käyttävät ng –nimiavaruutta, minkä ansiosta ne on helppo paikallistaa html-kielen joukosta (AngularJS, 2016).

3.3 Bootstrap

Bootstrap on alun perin Twitterin suunnittelijoiden vuonna 2010 luoma avoimen lähdekoodin kirjasto selainohjelmistojen käyttöliittymäratkaisujen luontiin. Tuolloin kirjasto kulki vielä nimellä Twitter Blueprint. Sitä hyödyntämällä voidaan luoda tyyliään ja ulkoasultaan yhdenmukaisia sovelluskäyttöliittymiä. Bootstrap tarjoaa kokoelman käyttövalmiita komponentteja, kuten navigointipalkkeja, lomakkeita sekä erilaisia pikakuvakkeita (Bootstrap, 2016).

Bootstrap –käyttäjä voi valita useasta pakettivaihtoehdosta haluamansa, sekä halutessaan muokata valmiita elementtejä (Lehdonvirta, P. & Korpela, J. K., 2012).

3.4 JSON

JSON on kieliriippumaton tapa vaihtaa tietoa selaimen ja palvelimen välillä. Siinä on kahdenlaisia rakenteita; avain-arvo –pareja (olioita) sekä arvoista koostettuja listoja (JSON.org, 1999). JSONin avulla olio muutetaan tekstimuotoiseksi serverin luettavaan muotoon. Monissa tapauksissa JSON korvaa nykyään XML:än helppolukuisuutensa ansiosta (Miettinen, M., 2013).

JSON-oliot ovat tekstimuotoisia avain-arvopareja aaltosulkein toisistaan eroteltuina. Luettavuuden lisäämiseksi avain-arvoparit esitetään usein pystysuorana luettelona (Kuva 1).

```
{
    "nimi1": arvo1,
    "nimi2": arvo2
}
```

Kuva 1. esimerkki JSON –tyylillä esitetystä avain-arvoparista (Freepascal käyttäjäyhteisö, 2015). Attribuutin nimi vasemmalla lainausmerkeissä, erotettuna attribuutin arvosta kaksoispisteellä.

Mikäli arvo-osuus koostuu useammasta kuin yhdestä arvosta (esimerkiksi taulukko), sen arvot esitetään hakasulkeiden sisällä pilkuilla toisistaan eroteltuina. Jos arvo-osuus taas on olio, sen arvot esitetään aaltosulkeissa kuten alla (Kuva 2) (Freepascal.org, 2016).


```
...{"address":{"street":"145 Koinange Street", "City":"Nairobi", "Country":"Kenya"}}
```

Kuva 2. esimerkki moniarvoisesta JSON-tyylillä esitetystä arvoparista (Freepascal käyttäjäyhteisö, 2015)

3.5 Ketterä ohjelmistokehitys ja Scrum

Ketterällä ohjelmistokehityksellä tarkoitetaan joukkoa menetelmiä, joita käytetään sovelluskehityksen projektinhallinnan välineinä. Niissä sovelluskehitys toteutetaan ennalta määrätyn pituisissa iteraatioissa, joissa kunkin päätteeksi tarkoituksena on tarjota asiakkaalle ominaisuuksiltaan toimiva versio sovelluksesta (Ite wiki crew, 2015). Ketterää ohjelmistokehitystä pidetään vaihtoehtona vanhemmalle vesiputousmallille, jossa sovelluskehitysprosessi nähdään elinkaarena, jolla on alku, välivaiheet ja loppu. Vesiputousmalli koetaan nykyisellään kömpelöksi ja joustamattomaksi ohjelmistokehitykseen sen lineaarisuuden takia; seuraavan vaiheen toteutusta ei voida aloittaa ennen edellisen vaiheen valmistumista (Taina, J., 2009).

Scrum on ketterässä ohjelmistokehityksessä käytetty projektinhallinnan tekniikka. Siinä sovelluskehitysjaksot jaetaan itsenäisiksi sprinteiksi, joista kunkin tavoitteena on ominaisuuksiltaan julkaisukelpoinen sovellusversio. Ennen kutakin sprintiä pidetään suunnittelupalaveri, ”sprint planning”, jossa sovitaan sprintin kehityskohteiksi otettavat toiminnallisuudet. Päivittäin pidetään päiväpalaverit, ”daily scrumit”, joissa sovitaan kunkin kehittäjän kyseisen päivän tehtävät. Jokaisen sprintin päätteeksi pidetään katselmus, ”sprint review”, jossa kyseisen sprintin aikaansaannokset käydään läpi ja esitellään asiakkaalle (Cohn, M., 2016).

3.6 Henkilötietojen käsittely

HeiHei Study & Travel –järjestelmä kerää siihen rekisteröityvistä opiskelijoista ja yrityksen työntekijöistä joitakin henkilötietoja, jotka tallennetaan tietokantaan. Henkilötietojen keräämisestä ja käsittelystä säädetään henkilötietolaissa, jota oli siis tarpeen noudattaa myös sovelluskehitystyössä.

Henkilötietolain mukaan henkilötietojen keräämisen täytyy aina olla perusteltua ja käyttötarkoitussidonnaista yrityksen toiminnan kannalta. Lain mukaan rekisteröidyn tulee myös antaa suostumuksensa henkilötietojen keräämiseen ja käsittelyyn (Henkilötietolaki, 1999). Henkilötietolain antamien rajoitteiden pohjalta järjestelmään kerättiin vain kontaktien ylläpidon kannalta tärkeimmät yhteystiedot (sähköpostiosoite, puhelinnumero) sekä kouluhakujen kannalta oleelliset henkilötiedot (nimi, koulutus, syntymävuosi).

Edellä mainitusta syystä esimerkiksi henkilötunnusta ei järjestelmään tallennettu, vaikka se olisikin kantasuunnittelussa voinut toimia erittäin hyvänä käyttäjät yksilöivänä tietona.

3.6.1 Henkilötietojen siirrosta ulkomaille

Koska HeiHei Study & Travel on kansainvälistä yhteistyötä tekevä yritys, täytyi tietojärjestelmän suunnittelussa ottaa huomioon myös henkilötietojen ulkomaille luovuttamista koskevat säädökset. Pääsääntönä voidaan pitää, että jotta henkilötietoja voidaan luovuttaa ulkomaille, vastaavan siirron tulee olla sallittua kyseisessä tilanteessa myös Suomessa. Mikäli luovutuksen vastapuolena on EU:hun kuulumaton maa, tulee siirron täyttää EU:n ulkopuolelle luovutettavia henkilötietoja koskevat erityisedellytykset, jotka on niin ikään määritelty henkilötietolaissa (Tietosuojavaluutettu, 2013).

Henkilötietojen siirrossa ulkomaille tulee aina varmistua siitä, että vastaanottajaosapuolella on oikeus kyseisten henkilötietojen käsittelyyn. Henkilötietojen lähettäjän tulee myös varmistaa, että tietojen vastaanottaja on ajan tasalla henkilötietojen käsittelyä koskevista säädöksistä ja myös noudattaa näitä säädöksiä (Tietosuojavaluutettu, 2013).

Henkilötietojen luovutuksesta EU:n sisällä on säädetty henkilötietodirektiivissä.

Henkilötiedoilla on EU:n sisällä siis vapaa liikkuvuus, eli niitä koskevat aina lähetysmaassa kulloinkin voimassa olevat säädökset. Lisäksi EU –maiden välisestä siirrosta ei tarvitse tehdä ilmoitusta tietosuojavaluutetulle (Tietosuojavaluutettu, 2013).

Mikäli tietoja ollaan luovuttamassa EU –alueen ulkopuolelle, on aina arvioitava, että vastaanottavan maan tietosuojataso on riittävän korkea. Tietosuojan tason arvioinnissa huomioon otettavia seikkoja ovat tietojen luonne, käsittelyn tarkoitus ja kesto aika, vastaanottajamaassa voimassa olevat yleiset ja alakohtaiset oikeussäännöt ja käytänteet sekä noudatettavat turvatoimet (Henkilötietolaki, 1999).

4 CRM-järjestelmän toteutus

Opinnäytetyön kohteena oli kehittää CRM -, eli asiakashallintajärjestelmä pienyrityksen tarpeisiin. Tarve järjestelmälle syntyi, koska asiakastietojen ja dokumenttien hallinta koettiin hankalaksi tietojen sijaitessa useammassa eri paikassa. Tavoitteena oli siis tietojen keskittäminen ja niiden helppokäyttöinen hallinta asiakasyrityksen palvelimelta käsin.

Käytetyt menetelmät määräytyivät palvelimen asettamien rajoitusten perusteella.

Asiakkaan käyttämä palvelin ei tukenut Javaa, joten luonnollinen valinta ohjelmointikieleksi oli PHP, koska siitä ohjelmointikokemusta löytyi jo.

Selainsovelluksessa tiedot esitettiin AngularJs –ohjelmistokehystä käyttäen.

Taustajärjestelmästä selaimelle tiedot siirrettiin JSON-tyylisesti esitettynä objektina, ja itse käyttöliittymä toteutettiin Bootstrap-valmiskirjaston pohjalta.

Bootstrap tarjoaa suuren määrän valmiita avoimen lähdekoodin käyttöliittymäkehyskiä, joista tämän sovelluksen tarpeita vastasi parhaiten 'Ruma Bootstrap' –pohja.

Valmiskomponenttien lisäksi selainsovellukseen toteutettiin joustavan käytön vaatimia yksilöllisiä muotoiluja. Lopputulemana syntyi yksisivuinen selainsovelluksen käyttöliittymä, johon toteutettiin omat näkymät peruskäyttäjille ja pääkäyttäjälle.

4.1 Järjestelmän rajoitusten kartoitus

Järjestelmän käyttäjäryhmiksi valittiin admin –käyttäjät (HeiHei Travelin työntekijät) sekä opiskelijat. Molemmat rekisteröityvät järjestelmään käyttöliittymässä olevan rekisteröitymislomakkeen kautta, mutta admin-oikeudet pääkäyttäjille asetetaan palvelimelta käsin.

Opiskelijoiden kouluhaut tapahtuvat järjestelmästä käsin, kuten myös hakujen poisto tarpeen vaatiessa.

Opiskelijat lataavat järjestelmään hakuprosessiin liittyviä dokumentteja ja seuraavat samalla hakunsa etenemistä. He eivät voi kuitenkaan muuttaa oman edistymisprosessinsa tietoja. Jokaisella opiskelijalla on henkilökohtainen checkbox-lista, josta voi nähdä hakuprosessin vaatimat tiedot ja seurata sen etenemistä.

Pääkäyttäjät voivat tarkastella ja muuttaa kaikkien opiskelijoiden hakutietoja, lisätä ja poistaa haettavia kouluja ja päivittää niihin liittyvää dokumentaatiota. Admin-käyttäjien käyttövaltuuksiin ei kuulu opiskelijoiden henkilökohtaisten tietojen (nimi, osoite jne.) muuttaminen. Tehtiin myös päätös, että järjestelmän tässä versiossa kummankaan käyttäjäryhmän käyttäjät eivät vielä voi päivittää rekisteröitymisen yhteydessä annettuja

henkilökohtaisia tietoja selainsovelluksesta käsin. Näiden tietojen muuttaminen tai päivitys luonnollisesti onnistuu palvelimella.

4.2 Tietotarpeiden kerääminen

Lista tietotarpeista saatiin toimeksiantajalta. Keskeisimmässä roolissa olivat checkbox-listaan tarvittavat tiedot, joissa oli paljon koulukohtaista vaihtelua. Tiedot lisättiin kunkin koulun checkbox-listaan, jota pääkäyttäjät voivat myöhemmin tarvittaessa muuttaa.

4.3 Käyttöliittymän suunnittelu

Käyttöliittymä suunniteltiin tietotarpeiden pohjalta; niiden perusteella suunniteltiin ensimmäiset hahmotelmat selainsovelluksen sivuista. Hahmotelmien perusteella valittiin tarpeisiin sopiva Twitter Bootstrap –pohja ja aloitettiin web-sivujen luonti. Suunnittelun ja asiakastoiveiden perusteella suuri osa tiedoista päädyttiin esittämään listana.

Jokaisen sprintin jälkeen pidetyssä Sprint Reviewissä asiakkaalle demottiin sprintin aikaansaannokset, ja heillä oli mahdollisuus antaa palautetta ja kehitysehdotuksia käyttöliittymän ulkoasuun.

Selainsovelluksen käyttäjäryhmiksi valittiin admin-käyttäjät vastaamaan HeiHeiTravelin – henkilökunnan ylläpitotarpeisiin, sekä opiskelija-käyttäjät yrityksen asiakkaita varten. Suunnittelussa otettiin huomioon useimmin käytettävät tietosisällöt sekä niiden looginen esittäminen molemmille käyttäjäryhmille.

Admin –käyttäjällä on oman profiilinsa lisäksi pääsy listauksiin kaikista opiskelijoista, näiden hakutilanteesta ja yhteistyökouluista. Admin –käyttäjät voivat myös päivittää edellä mainittuja listoja, sekä ladata tiedostoja erikseen joko kouluille tai käyttäjille.

Opiskelija-käyttäjät pääsevät tarkastelemaan ja päivittämään omaa profiiliaan ja hakutilannettaan, sekä lisäämään ja poistamaan kouluja hakulistaltaan.

4.4 Selainsovelluksen rakenne ja toteutus

Selainsovelluksen toteutus aloitettiin valitsemalla soveltuva Twitter Bootstrap –pohja, sekä luomalla raakaversiot yksittäisistä html-sivuista. Sisäänkirjautumis- ja rekisteröitymissivu luotiin hyödyntämällä valmispohjaa tyylikkään ja yhtenäisen ulkonäön saavuttamiseksi (Kuva 3).

HeiHei Travel - Rekisteröidy tai kirjaudu sisään

Rekisteröidy tai kirjaudu sisään

Kirjaudu sisään

Täytä allaoleva lomake rekisteröityäksesi:

Etunimi...

Sukunimi...

Email...

Salasana...

Vahvista salasana...

Puh...

Tutkinto...

Paikkakunta...

Kuva 3. 'Ruma Bootstrap'-pohjalla luotu rekisteröitymislomake

5 Käyttäjäpolut

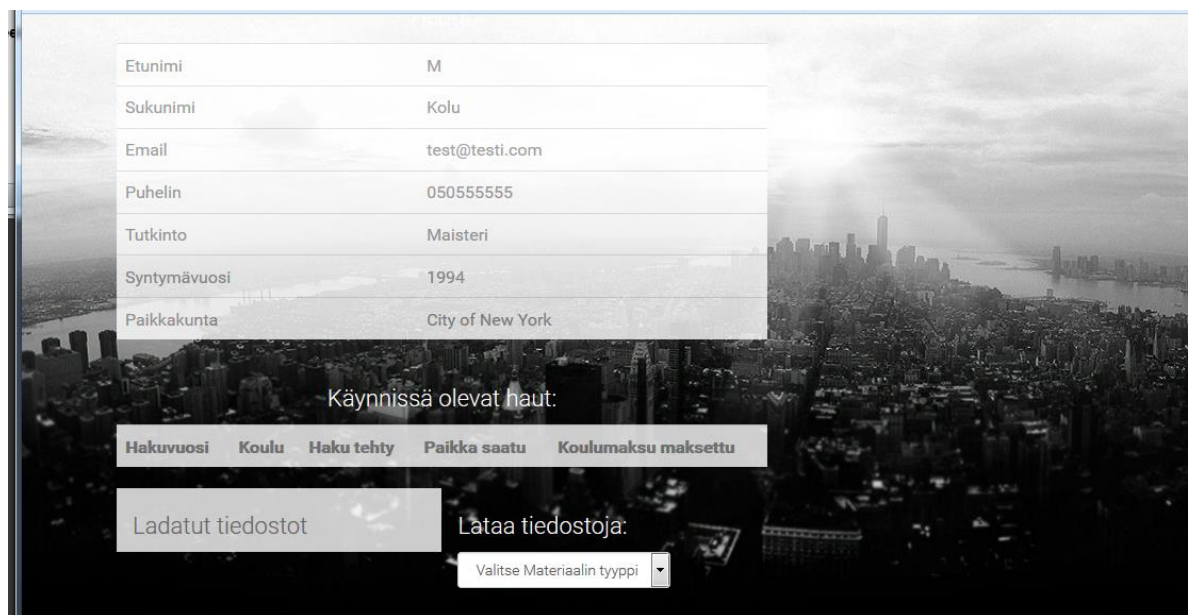
Tässä kappaleessa on esitelty esimerkkitapaukset molempien käyttäjäryhmien yksittäisistä käyttäjäpoluista

5.1.1 Admin-käyttäjän polku

'Admin' -käyttäjän rekisteröitymisen yhteydessä tämä ohjataan etusivulle, josta tämä voi lukea järjestelmän käyttöehdot. Myöhemmillä sisäänkirjautumiskerroilla käyttäjä ohjataan suoraan omalle profiilisivulleen.

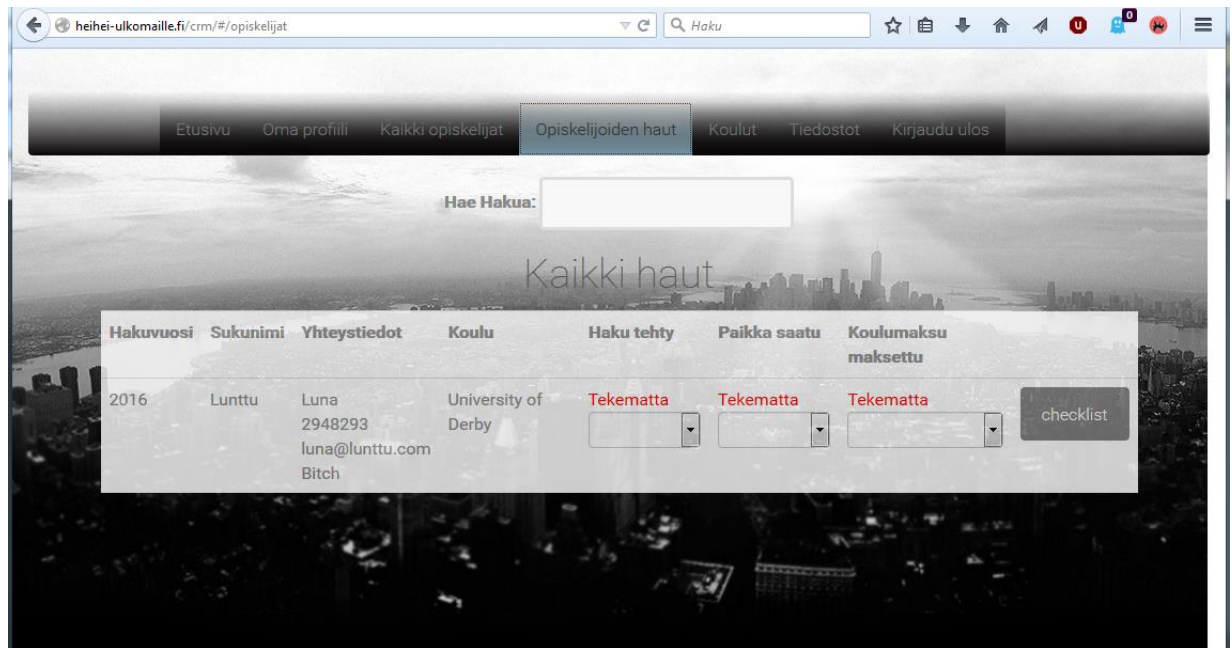
'Admin' -käyttäjä pääsee oman profiilinsa lisäksi tarkastelemaan kaikkien opiskelijoiden profiileita, kaikkia käynnissä olevia hakuja, kouluja ja niihin liittyviä materiaaleja sekä ladata tiedostoja kouluille tai käyttäjille.

Opiskelijoiden profiilit on esitetty nappeina, joita painamalla saa esiin valitun opiskelijan tiedot (Kuva 4). Tähän vaihtoehtoon päädyttiin lisäämään myös mahdollisuus ladata puuttuvia dokumentteja opiskelijakohtaisesti.

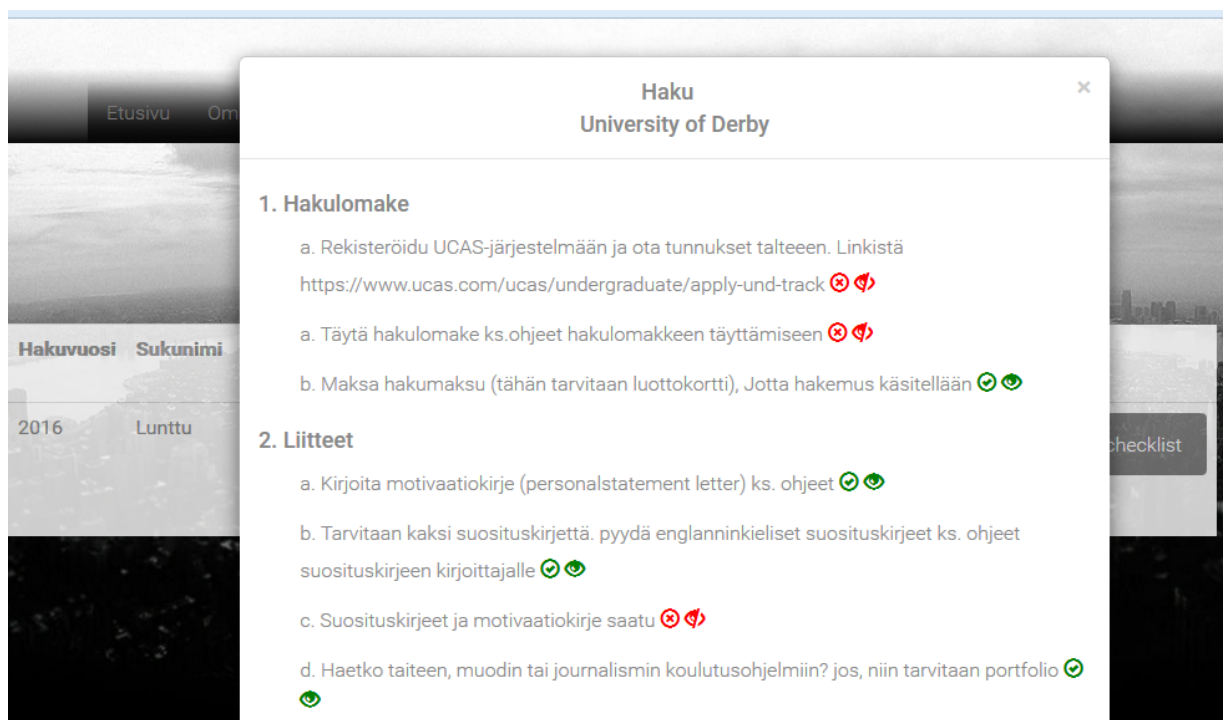


Kuva 4. Avatun opiskelijan profiilin tiedot

Opiskelijoiden haut -välilehden alta löytyvät kaikkien opiskelijoiden haut eriteltyinä (Kuva 5). Lisäksi sivulla on hakukenttä, jossa yksittäistä opiskelijan kouluhakua voi hakea esimerkiksi opiskelijan nimen perusteella, tai vaihtoehtoisesti hakea kaikki johonkin tiettyyn kouluun kohdistuvat haut. Lisäksi kunkin haun etenemisen tilannetta voi muuttaa, sekä hakukohtaisen checklist-painikkeen alta voi tarkastaa tai muuttaa, mitkä hakuun tarvittavat asiakirjat opiskelija on jo toimittanut (Kuva 6).

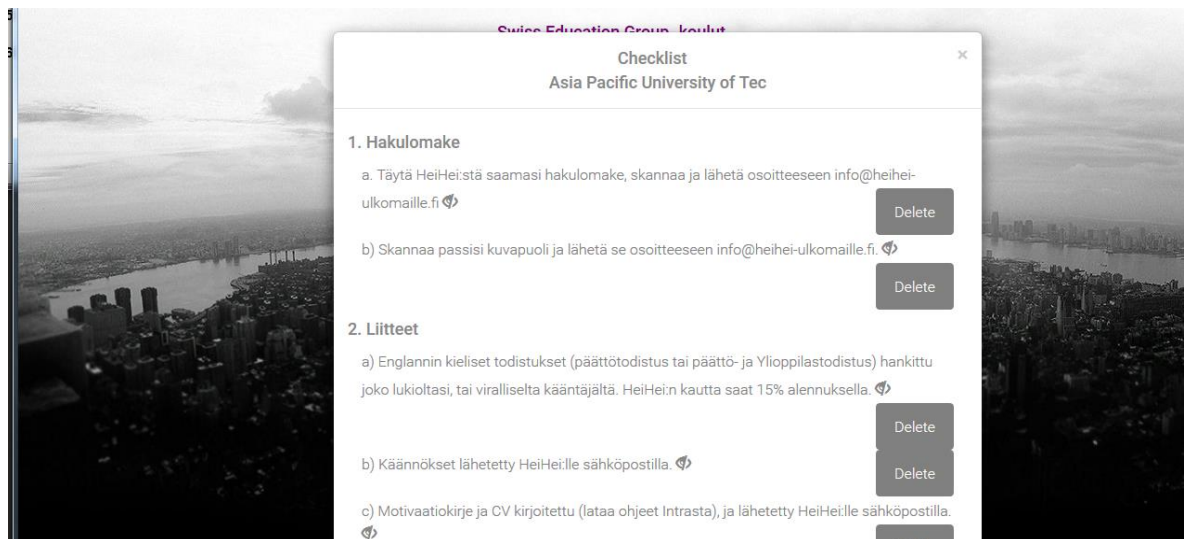


Kuva 5. Opiskelijoiden haut-välilehden tiedot

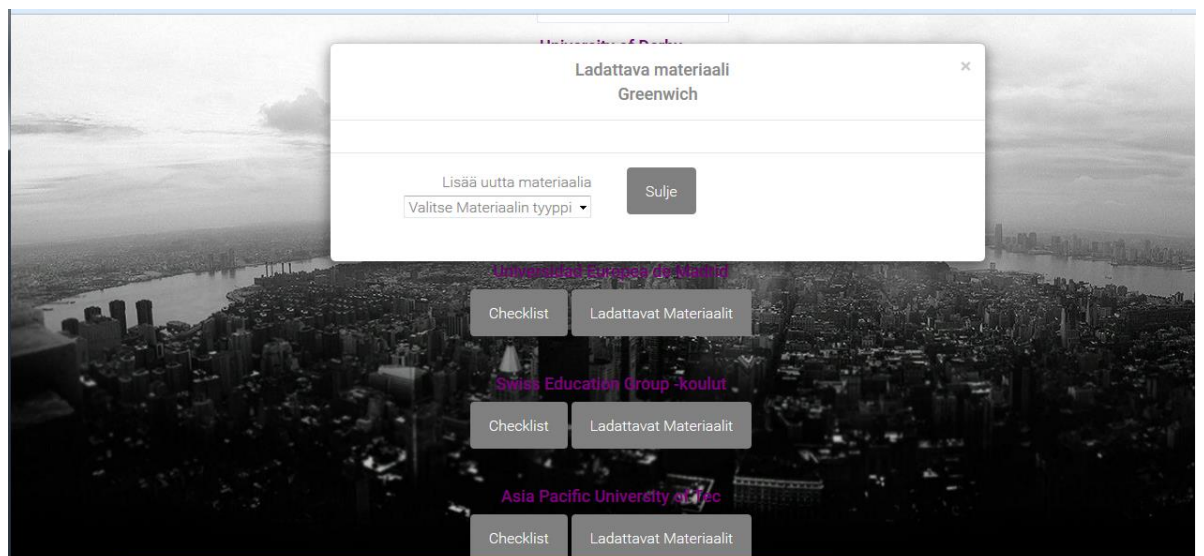


Kuva 6. Checklist-listaus

Koulut –välilehden alla on lista kullakin hetkellä hakuvaihtoehtoina olevista kouluista ja niihin liittyvistä hakumateriaaleista. Admin-käyttäjälle näkyvät koulukohtaiset checklist-, ja ladattavat materiaalit-painikkeet, sekä sivun alalaidasta löytyvä lisää uusi koulu –painike. Checklist-välilehden alla näkyvät hakumateriaalit, ja ne voi poistaa tai niiden näkyvyyttä opiskelijoille voi säätää tätä kautta (Kuva 7). Ladattavat materiaalit-painikkeen takaa kullekin koululle voi lisätä uutta materiaalia checklistiin (Kuva 8). Lisää uusi koulu-painike on kokonaan uuden koulun lisäämistä varten.

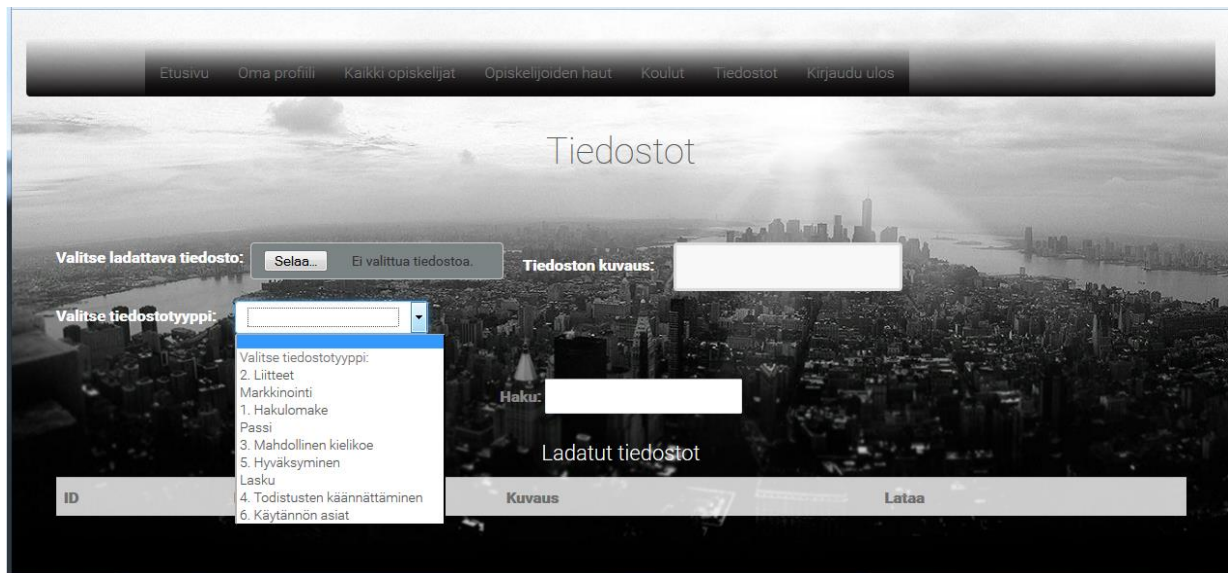


Kuva 7. Koulukohtainen checklist-hallintapaneeli admin-käyttäjälle



Kuva 8. Koulukohtaisen materiaalin lisäys admin-käyttäjänä

Tiedostot –välilehden takaa löytyy lista ladatuista tiedostoista, sekä mahdollisuus ladata lisää materiaalia järjestelmään (Kuva 9). Nämä tiedostot ovat hallinnollisia, ja ovat vain HeiHeiTravelin työntekijöiden käytössä.

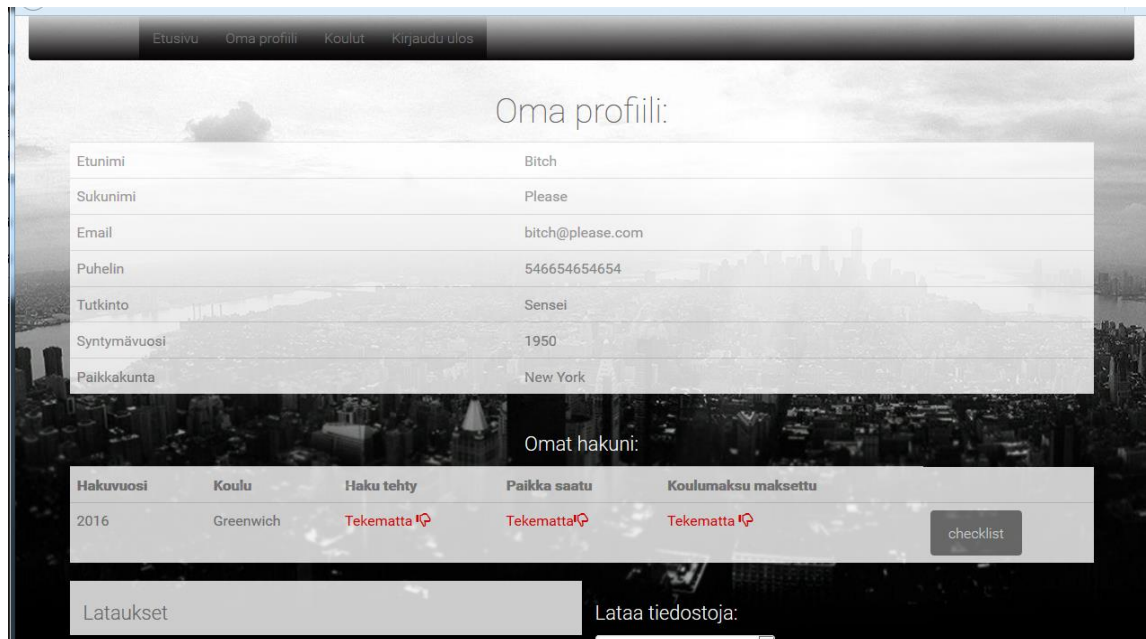


Kuva 9. Admin -käyttäjän tiedostojen latausvälilehti

5.1.2 Opiskelijakäyttäjän polku

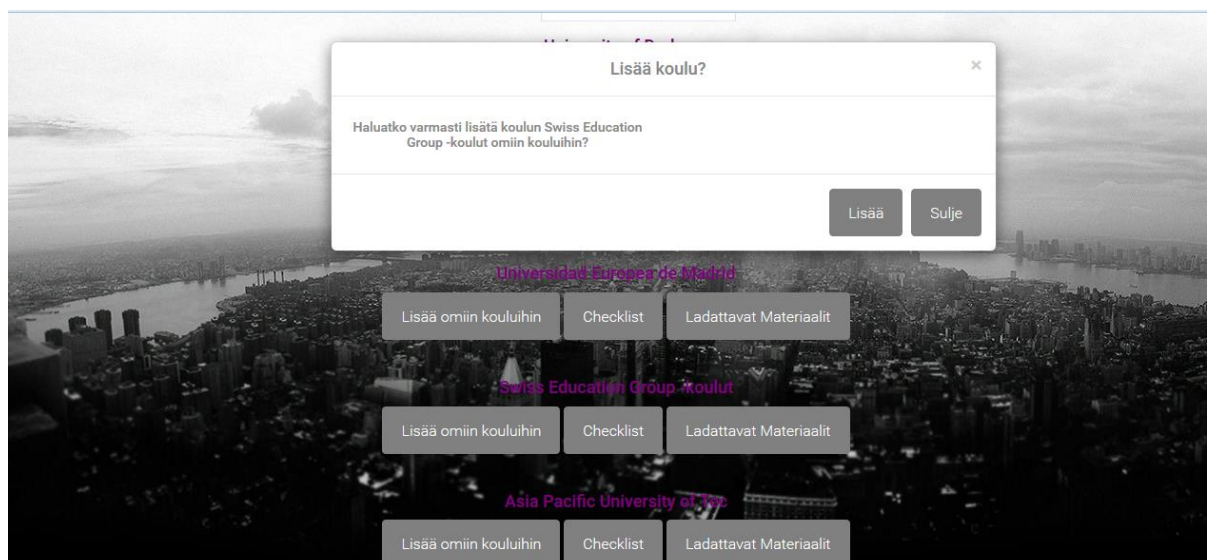
Kun opiskelijakäyttäjä lataa ensimmäisen kerran crm-järjestelmän etusivun, hänelle annetaan vaihtoehtoisiksi joko rekisteröityminen tai sisäänkirjautuminen. Rekisteröitymisen yhteydessä opiskelija antaa järjestelmän tarvitsemat henkilökohtaiset tiedot (nimi, puhelinnumero, sähköpostiosoite, tutkinto, paikkakunta, syntymävuosi ja salasana), jonka jälkeen tämä kirjataan suoraan sisään järjestelmään. Jatkossa käyttäjätunnuksena toimii annettu sähköpostiosoite.

Opiskelija käyttäjä ohjataan ensimmäisen sisäänkirjautumisen yhteydessä etusivuvälilehdelle, jossa tiedot järjestelmän käyttöehdoista ovat luettavissa. Myöhemmillä sisäänkirjautumiskerroilla opiskelija-käyttäjä ohjataan suoraan 'oma-profiili' -välilehdelle. Oma profiili -välilehdeltä löytyvät opiskelijan henkilökohtaiset tiedot sekä tiedot eri hakuprosessien etenemisestä (Kuva 10). Hakukohtaisen checklist -näppäimen takaa löytyy myös listaus toimitettavista dokumenteista sekä merkintä siitä, onko opiskelija jo toimittanut ko. dokumentin. Näitä dokumentteja voi ladata sivun alalaidassa olevan lataa dokumentteja-painikkeen avulla.



Kuva 10. Opiskelijakäyttäjän 'oma profiili' -välilehti

Oletetaan, että opiskelija haluaa toteuttaa uuden kouluhaun. Tämä tapahtuu 'koulut' – välilehden kautta. 'Koulut'-välilehdellä opiskelijalle näkyvät kaikki HeiHeiTravelin yhteistyökoulut, sekä jokaiselle koululle vaihtoehdot 'Lisää omiin kouluihin', 'checklist' ja 'ladattavat materiaalit'. Opiskelijan halutessa lisätä koulun omiin hakuihinsa hän painaa 'lisää omiin kouluihin' –painiketta, ja vastaa varmistuskysymykseen myöntävästi (Kuva 11). Tämän jälkeen kyseinen koulu näkyy 'oma profiili' –välilehden 'omat hakuni'-kohdassa, ja 'Koulut'-välilehdellä koulun kohdalta poistuu 'lisää omiin kouluihin'-painike päällekkäisten hakujen estämiseksi.



Kuva 11. Koulun lisääminen omiin kouluihin

Lisättyään koulun omiin kouluihinsa opiskelija voi tarkastaa koulun vaatimat materiaalit 'Koulut'-välilehden 'checklist'-painikkeen takaa. Mikäli opiskelijalla on jo valmiina jokin

koulun vaatimista materiaaleista, hän voi ladata sen 'Oma profiili'-välilehden alalaidasta löytyvästä 'Lataa tiedostoja' -kohdasta. Ensin on valittava materiaalin tyyppi, jonka jälkeen aukeaa valintaikkuna josta halutun tiedoston voi valita ladattavaksi. Valittuaan tiedoston opiskelija voi kirjoittaa vielä tarkemman kuvauksen tiedoston sisällöstä, ja ladata sen palvelimelle painamalla 'lataa tiedosto'-painiketta. Tämän jälkeen ladattu tiedosto näkyy 'Oma profiili -välilehden 'lataukset' -kohdassa.

Jos latausten myötä opiskelijan hakuprosessi on edennyt riittävästi, opiskelija voi muuttaa haun etenemiseen liittyviä tietoja painamalla 'Omat hakuni' -alla kyseisen haun kohdalla olevaa 'checklist' -painiketta. Tämän jälkeen aukeaa lista hakuvaiheista, jotka lähtökohtaisesti ovat kaikki 'tekemättä' -tilassa. Jokaisen hakuvaiheen perässä on painike, joka ilmaisee vaiheen tilan (tekemättä/tehty). Klikkaamalla painiketta opiskelija voi muuttaa hakuvaiheen tilaa (Kuva 12). Muutokset tallentuvat automaattisesti kun opiskelija sulkee valintaikkunan.

Etusivu Oma profiili Koulut

Etunimi

Sukunimi

Email

Puhelin

Tutkinto

Syntymävuosi

Paikkakunta

Hakuvuosi **Koulu**

2016 Greenwich

Lataukset

survey.txt sdfsd

Download

Haku Greenwich

1. Hakulomake

a. Täytä hakulomake ks.ohjeet hakulomakkeen täyttämiseen ✓

b. Täytä hakulomake ks. ohjeet hakulomakkeen täyttämiseen ✓

c. Maksa hakumaksu (tähän tarvitaan luottokortti), jotta hakemus käsitellään ✓

2. Liitteet

a. Pyydä englanninkielinen suosituskirje opettajaltasi ks.ohjeet suosituskirjeen kirjoittajalle ja toimita se HeiHei:lle ✗

3. Mahdollinen kielikoe

a. Varaa IELTS-kielikoe <https://www.finnbrit.fi/exams-tests> (mikäli pitkän englannin ylioppilastutkinnon arvosanasi on M tai parempi, ei sinun tarvitse tehdä kielikoetta) ✗

b. Toimita kielikokeen tulos kirjeitse Greenwichiin. Osoitteen saat HeiHei:ltä. ✗

4. Todistusten käännättäminen

a. Toimita suomenkielinen todistus HeiHei:lle ✗

c. Maksa todistuskäännös. Kääntäjä lähettää maksulinkin sähköpostiisi. ✗

d. Toimita englanninkielinen todistus HeiHei:lle ✗

5. Hyväksyminen

a. Tarjous paikoista saatu ✗

b. Vähintään yksi tarjous on hyväksytty (toinen varavaihtoehto mahdollista valita lisäksi). ✗

c. e. d. c. Allekirjoita ja palauta acceptance of place yliopistolle ✗

d. Maksa pre-registration deposit ✗

Kuva 12. Opiskelijan 'omat hakuni' -checklist

5.2 AngularJS –integraatio

Ylläolevat kuvat ovat järjestelmän viimeisestä versiosta, jolloin integraatio backend – puolen kanssa oli jo toteutettu. PHP-koodit ovat omissa tiedostoissaan, ja niiden avulla kannasta haetut tiedot esitetään selainjärjestelmässä AngularJS:ää hyödyntäen. HTML-sivujen muuttuva sisältö (opiskelijat, haut, koulut) toteutettiin taulukoina tai nappeina, jolloin dynaaminen sisältö oli joustavasti generoitavissa elementteihin silmukkaa käyttäen.

Muuttuvan sisällön liittäminen onnistui lisäämällä haluttuihin HTML-elementteihin AngularJS:n ng-tagit. Esimerkiksi silmukan ollessa kyseessä käytettiin ng-repeat –tagia. Opiskelija-välilehdellä opiskelijat olivat esitetty nappeina, joita painamalla sai kunkin opiskelijan henkilökohtaiset tiedot esille. Tämä onnistui liittämällä napin yhteyteen ng-show –tagin.

'Checklist' –painikkeista aukeavat ikkunat toteutettiin käyttämällä Bootstrapin 'modal' –elementtiä, ja niihin haettiin halutut tiedot taustajärjestelmästä käyttämällä niinikään AngularJS:n ng-repeat –toimintoa sekä Bootstrapin 'data-target' –elementtiä.

5.3 Uudelleenmäärittely ja muutosten toteutus

Asiakkaan kanssa pidetyssä loppupalaverissa käytiin järjestelmä toimintoineen läpi, sekä kirjattiin toivotut muutosehdotukset. Aikataulullisista syistä kaikkia muutosehdotuksia ei ollut mahdollista enää toteuttaa, kuten esimerkiksi sitä, että opiskelijat olisivat itse voineet muuttaa hakuprosessin tilaa. Sen sijaan käyttöliittymään toteutettiin muutamia selkeyden kannalta olennaisia muutoksia, kuten esimerkiksi poistettiin turhaan kahdessa paikassa näkyviä samoja tietoja yksittäisen opiskelijan hakuprosessin etenemisestä.

6 Yhteenveto

Opinnäyteprosessin päämääränä oli luoda yksisivuinen CRM-järjestelmä pienyrityksen tarpeisiin asiakasyrityksen toiveiden pohjalta. Alun perin aikaa projektiin oli varattu neljä kuukautta, syyskuun alusta joulukuun loppuun 2015. Järjestelmän laajuudesta sekä kasvaneista asiakasvaatimuksista johtuen aikataulua jouduttiin kuitenkin pidentämään helmikuulle 2016. Tähän ratkaisuun päädyttiin yhdessä asiakasyrityksen kanssa, sillä molemmille osapuolille oli kannattavampaa luoda käytännöllinen ja ominaisuuksiltaan toimiva järjestelmä.

Asiakkaan toivomat ominaisuudet saatiin toteutettua, mutta projektin edetessä hankaluudeksi koettiin vaatimusten muuttuminen matkan varrella. Ongelman ratkaisuksi riittänee se, että asiakasosapuoli toteuttaa omakohtaisen vaatimusmäärityslistan; tällä kertaa vaatimusmäärittelyt kirjoitettiin sovelluskehittäjien toimesta asiakkaan kanssa käytyjen Skype-palaverien yhteydessä.

Vaikka kaikki toivotut ominaisuudet saatiin toteutettua, oli projektin aikataulu lopulta niin tiukka, ettei viimeistelylle jäänyt kunnolla aikaa. Selainjärjestelmän elementit ovat pääpiirteittäin kohdillaan, mutta dynaamisen sisällön astuessa mukaan kuvaan jotkut elementit käyttäytyvät ei-toivotulla tavalla. Jatkokehityskohteeksi jää siis ainakin selainjärjestelmän ulkoasun siistiminen ja viimeisteleminen käyttäjäystävällisempään muotoon.

AngularJS:n ja PHP:n yhteiskäyttö sujui ongelmitta, mutta AngularJS:n yhteensovittaminen Bootstrap –komponenttien kanssa asetti omat haasteensa. Tämäkin ongelma on ratkaistavissa joko pidentämällä projektiin varattua aikataulua tai kasvattamalla sovelluskehitystiimin kokoa. Loppupäätelmänä voikin todeta, että CRM-järjestelmä on hyvin laaja kokonaisuus kahden hengen tiimin toteutettavaksi, koska huomioon on otettava molempien eri käyttäjäryhmien käyttötarpeet ja nämä täyttävät ominaisuudet. Esimerkiksi 'Asiakas'-käyttäjä voi kokea hyödylliseksi mahdollisuuden muuttaa oman hakunsa etenemistä sekä poistaa niin sanottuja turhia hakuja; 'admin' –käyttäjälle edellä mainittu saattaa muodostua ongelmaksi jos hakuja päivitetään esimerkiksi etukäteen tai reaailtilanteen vastaisesti. Tämä asettaa omat haasteensa käyttöliittymäsuunnitteluun ja sen toteuttamiseen kaikki käyttäjäryhmät huomioivalla tavalla.

Lähteet

Austin, A., 2014, An Overview of AngularJS for Managers. Andrew Austin. Available at: <http://andrewaustin.com/an-overview-of-angularjs-for-managers/> [Accessed: 19 May 2016].

Bootstrap, 2016, Bootstrap: About [Online]. Available at: <http://getbootstrap.com/about/> [Accessed: 12 April 2016].

CodeProject, 2016, Developing a Large Scale Application with a Single Page Application (SPA) Using AngularJS [Online]. Available at: <http://www.codeproject.com/Articles/808213/Developing-a-Large-Scale-Application-with-a-Single> [Accessed: 19 May 2016].

Cohn, M., 2016, Scrum Frequently Asked Questions: Scrum Development: What's Involved? [Online]. Available at: <https://www.mountaingoatsoftware.com/agile/scrum#tab-232> [Accessed: 20 April 2016].

Freepascal käyttäjäyhteisö, 2015, JSON/fi [Online]. Available at: <http://wiki.freepascal.org/JSON/fi> [Accessed: 20 April 2016].

GitHub, 2016, AngularJS Changelog [Online]. Available at: <https://github.com/angular/angular.js/blob/master/CHANGELOG.md> [Accessed: 19 May 2016].

Google Inc., 2016, AngularJS Miscellaneous: #FAQ [Online]. Available at: <https://docs.angularjs.org/misc/faq> [Accessed: 22 February 2016].

HeiHei! Study, Work & Travel, 2016, Mikä ihmeen Heihei! Study, Work & Travel? [Online]. Available at: <http://heihei-ulkomaille.fi/heihei/> [Accessed: 22 February 2016].

Henkilötietolaki, 1999, 2. luku – Henkilötietojen käsittelyä koskevat yleiset periaatteet [Online]. Available at: <http://www.finlex.fi/fi/laki/ajantasa/1999/19990523#L2> [Accessed: 23 May 2016].

Henkilötietolaki, 1999, 5. luku – Henkilötietojen siirto Euroopan unionin ulkopuolelle [Online]. Available at: <http://www.finlex.fi/fi/laki/ajantasa/1999/19990523#L5> [Accessed: 23 May 2016].

Ite wiki crew, 2015, Ketterät menetelmät, agile, LEAN ja scrum. itewiki IT Expertise Wiki. Available at: <http://www.itewiki.fi/opas/ketterat-menetelmat-agile-lean-ja-scrum/> [Accessed: 20 April 2016].

JSON.org, 1999, Introducing JSON [Online]. Available at: <http://www.json.org/> [Accessed: 20 April 2016].

Lehdonvirta, P., & Korpela, J., K., 2012, HTML5 –kirja: Twitter Bootstrap [Online]. Available at: <http://html5kirja.fi/2012/08/22/twitter-bootstrap/> [Accessed: 22 February 2016].

Miettinen, M., 2013, JSON kiteytys. Available at: <https://prezi.com/f8ki3a32cgjd/json-kiteytys/> [Accessed: 20 April 2016].

Taina, J., 2009, Ohjelmistoprosessit ja ohjelmistojen laatu, luku 6 (Suunnitelmakeskeiset prosessit). Available at: https://www.cs.helsinki.fi/u/taina/opol/k-2009/pdf/luku-6_2.pdf [Accessed: 20 April 2016].

Tietosuojavaltuutettu, 2013, Henkilötietojen siirto ulkomaille henkilötietolain mukaan [Online]. Available at: http://www.tietosuoja.fi/material/attachments/tietosuojavaltuutettu/tietosuojavaltuutetun_toimisto/opaat/5JpRhVW1x/HENKILOTIETOJEN_SIIRTO_ULKOMAILLE_HENKILOTIETOLAIN_MUKAAN._10.6.2014pdf.pdf [Accessed: 19 May 2016].

Tutorials4Geeks, 2016, Single Page Application Using AngularJS Tutorial [Online]. Available at: <https://tests4geeks.com/tutorials/single-page-application-using-angularjs-tutorial/> [Accessed: 19 May 2016].